(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 702 477 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
26.01.2005 Bulletin 2005/04

(51) Int Cl.[7]: **H04L 29/06**, H04L 12/66

(21) Application number: **95306466.4**

(22) Date of filing: **14.09.1995**

(54) **System for signatureless transmission and reception of data packets between computer networks**

System für signaturlose Übertragung und Empfang von Datenpaketen zwischen Computernetzwerken

Système pour la transmission et la réception sans signature de paquets de données entre réseaux d'ordinateurs

(84) Designated Contracting States:
DE FR GB NL SE

(30) Priority: **15.09.1994 US 306337**

(43) Date of publication of application:
20.03.1996 Bulletin 1996/12

(73) Proprietor: **SUN MICROSYSTEMS, INC.**
Mountain View, CA 94043 (US)

(72) Inventors:
- **Mulligan, Geoffrey**
  Fremont, California 94555 (US)
- **Patterson, Martin**
  F-38000 Grenoble (FR)
- **Scott, Glenn**
  Sunnyvale, California 94086 (US)
- **Aziz, Ashar**
  Fremont, California 94555 (US)

(74) Representative: **W.P. Thompson & Co.**
Coopers Building
Church Street
Liverpool L1 3AB (GB)

(56) References cited:
- **FORNE J ET AL: "HARDWARE IMPLEMENTATION OF A SECURE BRIDGE IN ETHERNET ENVIRONMENTS" PROCEEDINGS OF THE GLOBAL TELECOMMUNICATIONS CONFERENCE (GLOBECOM), HOUSTON, NOV. 29 - DEC. 2, 1993, vol. 1, 29 November 1993, pages 177-181, XP000428050 INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS**
- **O'HIGGINS B ET AL: "SECURING INFORMATION IN X.25 NETWORKS" COMMUNICATIONS: CONNECTING THE FUTURE, SAN DIEGO, DEC. 2 - 5, 1990, vol. 2, 2 December 1990, pages 1073-1078, XP000220997 INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS**
- **SHARP R L ET AL: "NETWORK SECURITY IN A HETEROGENEOUS ENVIRONMENT" AT & T TECHNICAL JOURNAL, vol. 73, no. 5, 1 September 1994, pages 52-59, XP000475911**

EP 0 702 477 B1

Description

[0001] The present invention relates to the field of secure transmission of data packets, and in particular to a new method and system tor automatically encrypting and decrypting data packets between sites on the Internet or other networks of computer networks.

[0002] It is becoming increasingly useful for businesses to transmit sensitive information via networks such as the Internet from one site to another, and concomitantly more urgent that such information be secured from uninvited eyes as it traverses the internetwork. At present, unsecured data is replicated at many sites in the process of being transmitted to a destination site, and trade secret or other private information, unless secured, is thereby made available to the public.

[0003] It is possible for a user at the sending host to encrypt the data to be sent, and to inform the user who is to receive the data of the encryption mechanism used, along with the key necessary to decrypt. However, this requires communication and coordinated effort on the parts of both the sending and receiving users, and often the users will not take the requisite trouble and the packets will go unencrypted.

[0004] Even when these packets are encrypted, the very fact of their being transmitted from user A to user B may be sensitive, and a system is needed that will also make this information private.

[0005] Figure 1 illustrates a network of computer networks, including networks N1, N2 and N3 interconnected via a public network 10 (such as the Internet). When network N1 is designed in conventional fashion, it includes several to many computers (hosts), such as host A and additional hosts 20 and 30. Likewise, network N2 includes host B and additional hosts 40 and 50, while network N3 includes hosts 60-90. There may be many hosts on each network, and many more individual networks than shown here. For examples of such networks, reference is directed to J. Postel, User Datagram Protocol, RFC 768, August 28, 1980; *http://www.rfc-editor.org/rfc/rfc768.txt;* Information Sciences Institute, Internet Protocol: DARPA Internet Program Protocol Specification, RFC 791, September 1981; *http://www.rfc-editor.org/rfc/rfc791.txt;* and T Socolofsky, A TCP/IP Tutorial, RFC 1180, January 1991 *http://www.rfc-editor.org/rfc/rfcll80.txt;*

[0006] When a user at host A wishes to send a file, email or the like to host B, the file is split into packets, each of which typically has a structure such as packet 400 shown in Figure 7, including data 410 and a header 420. For sending over the Internet, the header 420 will be an internet protocol (IP) header containing the address of the recipient (destination) host B. In conventional fashion, each data packet is routed via the internetwork 10 to the receiving network N2, and ultimately to the receiving host B.

[0007] As indicated above, even if the user at host A encrypts the file or data packets before sending, and user B is equipped with the necessary key to decrypt them, the identities of the sending and receiving hosts are easily discernible from the Internet Protocol (IP) addresses in the headers of the packets. Current internetworks do not provide an architecture or method for keeping this information private. More basically, they do not even provide a system for automatic encryption and decryption of data packets sent from one host to another.

[0008] It is already known to transmit and receive packets of data via an internetwork from a first host computer on a first computer network to a second host computer on a second computer network, the first and second computer networks including, respectively, first and second bridge computers, each of said first and second host computers and first and second bridge computers including a processor and a memory for storing instructions for execution by the processor, each of said first and second bridge computers further including memory storing at least one predetermined encryption/decryption mechanism and information identifying a predetermined plurality of host computers as hosts requiring security for packets transmitted between them, the method being carried out by means of the instructions stored in said respective memories. Reference is directed in this connection to a paper by Forne J. et al entitled: 'HARDWARE IMPLEMENTATION OF A SECURE BRIDGE IN ETHERNET ENVIRONMENTS' PROCEEDINGS OF THE GLOBAL TELECOMMUNICATIONS CONFERENCE (GLOBECOM), HOUSTON, Nov. 29 -DEC. 2 1993, vol. 1, 29 November 1993, pages 177-181, XP000428050 INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS.

[0009] In the systems disclosed in the latter reference, each bridge computer ("secure bridge") comprises a three-layer cryptographic protocol arranged into three functional blocks referred to as a Communications Block, a Cryptographical Block and an Administrative Block, respectively. The Communications Block is the interface between the network and the other blocks. The Cryptographic Block ciphers/deciphers frames on the network with sensitive information by means of an algorithm. It also detects errors and attacks and informs the Administrative Block. The Administrative Block controls the secure bridge and communicates with a supervision and Administrative Centre.

[0010] In accordance with a first aspect of the present invention there is provided a method of encrypting data packets transmitted via an internetwork from a source host computer on a first computer network having a bridge to a destination host computer that is potentially part of a second computer network, the method comprising :

receiving at the bridge a data packet from the source host computer in the first computer network for transmission to the destination host computer, the data packet including an original header section and an original data section,

the original header section including a source identifier identifying the source host computer and a destination identifying the destination identifier host computer;
determining in the bridge that the data packet should be encrypted based at least in part upon reference to at least one of the source and destination identifiers; and
encrypting the data packet to produce an encrypted data packet;

the method being characterized:
in that the encryption of the data packet includes the encryption of both the original data section and the original header section;
by the step of generating an encapsulation header and appending the encapsulation header to the encrypted data packet to create a modified data packet, the encapsulation header including an internetwork broadcast address associated with at least one of the first and second networks, whereby the modified data packet, has the encapsulation header and an encrypted section that includes encrypted versions of the original header section and the original data section.

[0011] In accordance with a second aspect of the present invention there is provided
a method of decrypting data packets, comprising:

receiving an encrypted data packet from a source for a destination;
determining that the data packet is encrypted; and
decrypting the data packet to produce a decrypted data packet, characterised in that:

the encrypted data packet is provided with an encapsulation header storing a sourceidentifier and a destination identifier, at least one of which is a broadcast address,
the determining step that the data packet is encrypted is carried out upon reference to at least one of the source and destination identifiers;
the encrypted data packet includes an encrypted header section and an encrypted data section, and
after decryption the decryption header section is used as the header for the decrypted data packet.

[0012] In accordance with a third aspect of the present invention there is provided a bridge for use in a computer network, the bridge being arranged to route packets transmitted from and received by the network and to automatically encrypt selected packets passing through the bridge, the packets having a header section and a data section, the bridge comprising:

an encryption mechanism for encrypting selected packets;
means for determining whether a selected packet passing through a bridge is to be encrypted based at least in part on at least one of a source and a destination identified in the header section of the selected packet;

characterized by:

the encryption mechanism being arranged to encrypt an entire packet including both the header section and the data section; and
means for appending an encapsulation header to the encrypted packets, the encapsulation header having a broadcast address for at least one of a source identifier and a destination identifier for the packet, wherein the broadcast address utilized in the encapsulation header is different than the correspondence source or destination identifier in the selected packet.

[0013] In accordance with a fourth aspect of the present invention there is provided a bridge for use in a computer network, the bridge being arranged to route packets transmitted from and received by the network and to automatically decrypt selected packages passing through the bridge, characterised in that the bridges adapted to decrypt data packets having an encapsulation header section and an encrypted section that includes an encrypted header and an encrypted data section,
means for determining whether a selected packet passing through the bridge is to be decrypted based at least in part on at least one of a source or a destination identified in the encapsulation header section of the selected packet, being a broadcast address;
a decryption mechanism for decrypting packets that are to be decrypted, wherein the decryption mechanism eliminates the encapsulation header, thereby forming a decrypted packet that has a header section and a data section; and
means for forwarding decrypted packets to the destination address identified in the decrypted packet.

[0014] By virtue of the present invention, all messages that are predetermined to require encryption, e.g. all messages from a given host A to another host B, are automatically encrypted, without any separate action. on the part of the user. In this way, no one on the public internetwork can determine the contents of the packets. If the encapsulation header utilizes the network IP source and destination addresses, with the source and destination host addresses encrypted, then the host identities are also concealed, and an intervening observer can discern only the networks' identities.

[0015] The encapsulation header may include a field with an identifier of the source tunneling bridge. This is particularly useful if more than one tunnelling bridge is to be used for a given network (each tunnelling bridge having different encryption requirements and information), and in this case the receiving tunnelling bridge decrypts the data packets according to locally stored information indicating the encryption type and decryption key for all packets coming from the source tunnelling bridge.

[0016] By way of example only, specific embodiments of the present invention will now be described, with reference to the accompanying drawings, in which:-

[0017] Figure 1 is a diagram of a network of computer networks in conjunction with which the system of the present invention may be used.

[0018] Figure 2 is a block diagram of a host computer A on computer network N1 shown in Figure 1.

[0019] Figure 3 is a diagram of a network of computer networks incorporating tunnelling bridges according to the present invention.

[0020] Figure 4 is a block diagram of several tunnelling bridges of the present invention in a network of computer networks N1-N3 as shown in Figure 3.

[0021] Figure 5 is a diagram of another configuration of networks incorporating tunnelling bridges according to the present invention.

[0022] Figure 6 is a flow chart illustrating the method of signatureless tunnelling of the present invention.

[0023] Figure 7 illustrates a conventional data structure for a data packet.

[0024] Figures 8-11 illustrate modified data structures for use in different embodiments of the system of the invention.

[0025] Figure 12 is a block diagram of a network of computer networks including two tunnelling bridges of the invention on a single computer network.

[0026] The system of the present invention is designed to be implemented in existing computer networks, and in the preferred embodiment uses the addition of a tunnelling bridge at junctions between local computer networks and public or larger-scale networks such as the Internet. The mechanisms for carrying out the method of the invention are implemented by computers acting as these tunnelling bridges, incorporating program instructions stored in memories of the tunnelling bridges and appropriate (standard) network connections and communications protocols.

[0027] Figure 3 shows a network 100 of networks N1, N2 and N3 according to the invention, where each network includes a tunneling bridge -- TB1, TB2 and TB3, respectively -- which intercepts all data packets from or to the respective networks. Networks N1-N3 may in other respects be identical to networks N1-N3 in conventional designs. In the following description, any references to networks N1-N3 or hosts A and B should be taken as referring to the configuration shown in Figure 3, unless specified otherwise.

[0028] In this system, there are several modes of operation, numbered and discussed below as modes 1, 2, 2A, 3 and 3A. Mode 1 uses the configuration of Figure 1, while the other modes all use the configuration of Figure 3. The features of the tunnelling bridges TB1 and TB2 (including their program instructions, actions taken, etc.) in modes 2-3A are, in mode 1, features of, respectively, hosts A and B.

[0029] Each of the tunnelling bridges TB1-TB3 is preferably implemented in a separate, conventional computer having a processor and a memory, as shown in Figure 4. The memory may be some combination of random-access-memory (RAM), read-only-memory (ROM), and other storage media, such as disk drives, CD-ROMs, etc. The program instructions for each of the bridges TB1-TB3 are stored in their respective memories, and are executed by their respective microprocessors. The method of the present invention is carried out by a combination of steps executed as necessary by each of the processors of the sending host A, the tunnelling bridges TB 1 and TB2, and the receiving host B.

[0030] Encryption of data is an important step in the overall method of the invention, but the particular encryption mechanism used is not critical. It is preferable to use a flexible, powerful encryption approach such as the Diffie-Hellman method (see W. Diffie and M. Hellman, "New Directions in Cryptography", IEEE Transactions of Information Theory, November 1976). (The use of encryption in connection with IP data transfers is discussed in some detail in applicant's copending patent application, "Method and Apparatus for Key-Management Scheme for Use With Internet Protocols at Site Firewalls" by A. Aziz, Ser. No. 08/258,344 filed June 10, 1994, to which application reference is hereby directed). However any encryption scheme that provides for eneryption by a first machine, which sends the data packets, and decryption by a receiving machine, will be appropriate.

[0031] Figure 6 illustrates an embodiment of the method of the invention, and commences with the generation of data packets at the sending host A. The user at host A enters conventional commands for transmitting a file or the like from host A to host B, and the host computer A carries out the standard procedures for breaking the file down into data

packets as in Figure 7, each including both the data 410 and a header 400. In the case of transmissions over the Internet, this will be the IP header. Though the current discussion will be directed in large part to IP-specificimplementations, it should be understood that any network protocol may be used in conjunction with the present invention.

[0032] At box 200, the user at host A (see Figures 3 and 6) enters the conventional command for sending the file, email, or the like to a recipient, and host A generates data packets for sending over the Internet in the normal fashion. Each data packet initially has a structure like that of data packet 400 shown in Figure 7, including a data field 410 and a header field 420. The header 420 includes the destination address, in this example the IP address of host B.

[0033] The data packets are transmitted by host A at box 210, again in conventional fashion. However, at box 220, each packet is intercepted by the tunnelling bridge TB 1 (see Figures 3 and 4), when any of modes 2, 2A, 3 or 3A is used (see discussion below). When mode 1 (described below) is used, steps 220 and 280 are omitted, since this mode does not use tunnelling bridges; instead, the actions taken by the tunnelling bridges in modes 2-3A are all accomplished by the source and destination hosts themselves in mode 1. Thus, in the following discussion, wherever TB1 or TB2 is mentioned, it should be understood that in the case of mode I, the same feature will be present in host A or host B, respectively.

[0034] Stored in the memory of TB 1 (or host A, for mode 1) is a look-up table (not separately shown) of the addresses of hosts, both on the local network N1 and on remote networks such as N2 and N3, and an indication for each network whether data packets from or to that host should be encrypted. For instance, in this case the hosts table of TB1 indicates that any messages sent from host A to host B should be encrypted. Thus, bridge TB 1 (or host A) looks up hosts A and B in its tables, and determines that the data packets to be transmitted must first be encrypted, as indicated at boxes 230 and 240 of Figure 6.

[0035] Alternatively, the table could store the *network* identifiers (e.g. broadcast addresses) of networks N1 and N2, indicating that anything sent from network N1 to network N2 is to be encrypted. In this case, the table need not list each host in each network, which makes the table smaller and easier to maintain.

[0036] If each *host* is listed, however, greater flexibility can be retained, since it may be that messages to or from particular hosts need or should not be encrypted. In an alternative embodiment, the look-up table lists the networks N1 and N2 as networks to and from which packets should be encrypted, and also includes a hosts section of the table indicating exceptions to the normal encryption rule for these networks. Thus, if networks N1 and N2 are listed in the look-up table, then packets travelling from N1 to N2 should normally be encrypted; however, if there is an "exceptions" subtable indicating that *no* packets from host A are to be encrypted, then the normal rule is superseded. The exceptions can, of course, go both ways: where the normal rule is that the packets for a given network pair should/should not be encrypted, and the exception is that for this given host (source or recipient) or host pair, the packet should not/should nonetheless be encrypted. In this embodiment, the small size and ease of maintenance of the *network* tables is by and large retained, while the flexibility of the *hosts* table is achieved.

[0037] If the data to be transmitted from host A to host B (or network N1 to network N2) should not be encrypted, then the method proceeds directly to step 270, and the packet in question is transmitted unencrypted to the destination, via the Internet (or other intervening network).

[0038] In this example, the packets are encrypted at box 250. This is carried out by the tunnelling bridge TB1, according to whichever predetermined encryption scheme was selected, the primary requirement being that of ensuring that TB2 is provided with the same encryption scheme so that it can decrypt the data packets. TB2 must also be provided in advance with the appropriate key or keys for decryption.

The Encapsulation Header

[0039] At box 260, an encapsulation header is appended to the encrypted data packet. This header can take one of several alternative forms, according to the requirements of the user. Several modes of packet modification can be accommodated using the same basic data structure (but with differences in the information that is appended in the encapsulation header), such as the following:

Mode	Appended information
1	Encryption key management information (itself unencrypted) New IP header including originally generated IP addresses of source and destination hosts (unencrypted)
2	Encryption key management information (in encrypted form) Tunnelling bridge identifier for sender (unencrypted) New IP header including broadcast addresses of source and destination networks (unencrypted)
2A	(Same as mode 2, but without the tunnelling bridge identifier.)
3	Encryption key management information (encrypted)

(continued)

Mode	Appended information
	Optional: tunnelling bridge identifier for sender (unencrypted) New IP header including originally generated IP addresses of source and destination hosts (unencrypted)
3A	(Same as mode 3, but without the tunnelling bridge identifier.)

[0040] Data structures for modes 1, 2 and 3 are depicted in Figures 8, 9 and 10, respectively, wherein like reference numerals indicate similar features, as described below. The data structure for mode 2A is illustrated in Figure 11, and mode 3A may use the data structure of Figure 8.

[0041] The data structure 402 for mode 1 is represented in Figure 8. The original data 410 and original header 420 are now encrypted, indicated as (410) and (420). Encryption key management information 440 is appended (in encrypted form) as part of the new encapsulation header 430, along with a new IP header 450, including the addresses of the source and destination hosts. The information 430 includes indicates which encryption scheme was used.

[0042] Key management information can include a variety of data, depending upon the key management and encryption schemes used. For instance, it would be appropriate to use applicant's Simple Key-Management for Internet Protocols (SKIP).

[0043] In Figures 7-11, the fields with reference numerals in parentheses are encrypted, and the other fields are unencrypted. Thus, in Figure 8, the original data field 410 and address field 420 are encrypted, while the new encapsulation header 430, including the key management information 440 and the IP header 450, is not encrypted.

[0044] In this embodiment, the tunnelling bridges TB 1 and TB2 might not be used at all, but rather the hosts A and B could include all the instructions, tables, etc. necessary to encrypt, decrypt, and determine which packets are to be encrypted and using which encryption scheme. Mode 1 allows any intervening observer to identify the source and destination hosts, and thus does not provide the highest level of security. It does, however, provide efficient and automatic encryption and decryption for data packets between hosts A and B, without the need for additional computers to serve as TB 1 and TB2.

[0045] Alternatively, in mode 1 field 440 could include the IP broadcast addresses of the source and destination *networks* (instead of that of the hosts themselves), and in addition may include a code in the encryption key management information indicating which encryption scheme was used. This information would then be used by an intercepting computer (such as a tunnelling bridge) on the destination network, which decrypts the data packet and sends it on to the destination host.

[0046] In mode 2, a data structure 404 is used, and includes a new encapsulation header 432. It includes key encryption management information 440, which is appended to the original data packet 400, and both are encrypted, resulting in encrypted fields (410), (420) and (440) shown in Figure 9. A new IP header 470 including the broadcast addresses of the source and destination *networks* (not the addresses of the hosts, as in field 450 in Figure 8) is appended. In addition, a tunnelling bridge identifier field 460 is appended as part of the encapsulation header 432. Here, fields 410, 420 and 440 in this embodiment are all encrypted, while fields 460 and 470 are not.

[0047] The tunnelling bridge identifier identifies the *source* tunnelling bridge, i.e. the tunnelling bridge at the network containing the host from which the packet was sent. The recipient tunnelling bridge contains a tunnelling bridge look-up table, indicating for each known tunnelling bridge any necessary information for decryption, most notably the decryption method and key.

[0048] An appropriate tunnelling bridge identifier might be a three-byte field, giving 2^{24} or over 16 million unique tunnelling bridge identifiers. An arbitrarily large number of individual runnelling bridges may each be given a unique identifier in this way, simply by making the field as large as necessary, and indeed the field may be of a user-selected arbitrarily variable size. If desired, a four-byte field can be used, which will accommodate over 4 billion tunnelling bridges, far exceeding present needs.

[0049] Using mode 2, any observer along the circuit taken by a given data packet can discern only the tunnelling bridge identifier and the IP broadcast addresses for the source and destination networks.

[0050] The IP broadcast address for the destination network will typically be something like "129.144.0.0". which represents a particular network (in this case, "Eng.Sun.COM") but not any specific host. Thus, at intermediate points on the route of the packet, it can be discerned that a message is traveling from, say, "washington.edu" to "Eng.Sun.COM", and the identification number of the receiving tunnelling bridge can be determined, but that is the extent of it; the source and destination hosts, the key management information, and the contents of the data packet are all hidden.

[0051] Mode 2A uses the data structure shown in Figure 11, wherein the IP broadcast addresses for the source and recipient networks N1 and N2 are included in the encapsulation header field 470, but no tunnelling bridge identifier is used. This embodiment is particularly suitable for networks where there is only one tunnelling bridge for the entire network, or indeed for several networks, as illustrated in Figure 5.

[0052] In Figure 5, a packet sent from host C to host D will first be sent from network N4 to network N5, and will then be intercepted by the tunnelling bridge TB4, which intercepts all messages entering or leaving these two networks. TB4 will encrypt the packet or not, as indicated by its hosts look-up table. The packet traverses the public network and is routed to network N7, first being intercepted by tunnelling bridge TB5 (which intercepts all messages entering or leaving networks N6-N8), and at that point being decrypted if necessary.

[0053] In this embodiment or any embodiment where a packet is sent from a host on a network where a single tunnelling bridge is used for the entire source network or for multiple networks which include the source network, a tunnelling bridge identifier is not a necessary field in the encapsulation header. Since in this case only a given tunnelling bridge could have intercepted packets from a given host (e.g., TB4 for host C in Figure 5), the identity of the source tunnelling bridge is unambiguous, and the destination tunnelling bridge TB5 will include a table of hosts and/or networks cross-correlated with TB4. Having determined that tunnelling bridge TB4 was the source tunnelling bridge, TB5 then proceeds with the correct decryption.

[0054] This approach has certain advantages, namely that it eliminates the need to "name" or number tunnelling bridges, and reduces the sizes of the data packets by eliminating a field. However, a tunnelling bridge identifier field provides flexibility. For instance, in Figure 12, subnetworks N11 and N12 are part of one larger network N10, and each subnetwork N11 and N12 has its own assigned tunnelling bridge (TB7 and TB8, respectively). Thus, subnetworks N11 and N12 can be subjected to different types of encryption, automatically, and that encryption can be altered at will for one subnetwork, without altering it for the other.

[0055] A packet traveling from host F to host E in Figure 12 will include a source tunnelling bridge identifier (TB7) so that, when it reaches TB6 at network N9, it is identified correctly as having been encrypted by TB7 and not TB8. In this way, tunnelling bridge TB6 need maintain a table only the information pertaining to the tunnelling bridges, and does not need to maintain encryption/decryption specifics for the host or network level. (Note that TB6 still maintains information relating to *whether* to encrypt messages sent between host A and host B or network N1 and network N2, as the case may be, as discussed above.)

[0056] The tunnelling bridge identifier may be used for a variety of other purposes relating to the source tunnelling bridge, such as statistics recording the number of packets received from that tunnelling bridge, their dates and times of transmission, sizes of packets, etc.

[0057] An alternative to the use of hosts or networks tables in the memories of the source and destination tunnelling bridges (or source and destination hosts, as the case may be) would be any information identifying one or more predetermined criteria by which the source host or source tunnelling bridge determines whether to encrypt a given data packet Such criteria need not merely be source and destination information, but could include packet contents, time of transmission, subject header information, user id., presence of a key word (such as "encrypt") in the body of the packet, or other criteria.

[0058] Mode 3 uses a data structure 406 as shown in Figure 10, which is identical to the data structure 402 except for the addition of field 460 containing the tunnelling bridge identifier, which is the same as the tunnelling bridge identifier discussed above relative it mode 2.

[0059] In this embodiment, as in mode 1, field 450 includes the original host IP addresses for the source and destination *hosts* (not the addresses of the *networks*, as in mode 2), and thus an observer of a mode 3 packet will be able to determine both the original sender of the data packet and the intended receiver. Either mode contains sufficient information to route packets through an internet to a recipient network's tunnelling bridge for decryption and ultimate delivery to the recipient host.

[0060] Mode 3A may use the data structure shown in Figure 8, in conjunction with a network configuration such as those shown in Figures 3 or 12. The mechanisms and relative advantages are identical to those described above for mode 2A, while the structure reveals the source and destination host addresses.

[0061] Whichever encapsulation header is added at box 260 (see Figure 6), the packet is, at box 270, then transmitted to the destination network. At box 280, the destination network's tunnelling bridge (here, TB2 shown in Figure 3) intercepts the packet, which is accomplished by an instruction routine by which all packets are intercepted and inspected for encapsulation header information indicating encryption.

[0062] Thus, at box 290, the encapsulation header of the packet is read, and at box 300 it is determined whether the packet was encrypted. If a tunnelling bridge identifier forms a part of the encapsulated packet, then the method of encryption and decryption key are determined from the destination tunnelling bridge's (or destination host's, in the case of mode 1) local tables.

[0063] If no encryption was carried out on the packet, then it is sent on without further action to the correct host, as indicated at box 340. Otherwise, its encryption method is determined (box 320), and the packet is decrypted accordingly (box 330), and then sent on as in box 340.

Claims

1. A method of encrypting data packets transmitted via an internetwork from a source host computer (A) on a first computer network (N1) having a bridge (TB1) to a destination host computer (B) that is potentially part of a second computer network (N2), the method comprising :

   receiving at the bridge a data packet (400) from the source host computer (A) in the first computer network (N 1) for transmission to the destination host computer (B), the data packet (400) including an original header section (420) and an original data section (410), the original header section (420) including a source identifier identifying the source host computer (A) and a destination identifier identifying the destination host computer;
   determining in the bridge (TB1) that the data packet (400) should be encrypted based at least in part upon reference to at least one of the source and destination identifiers; and
   encrypting the data packet (400) to produce an encrypted data packet;

   the method being **characterized:**

   in that the encryption of the data packet includes the encryption of both the original data section (410) and the original header section (420);
   by the step of generating an encapsulation header (430, 432, 434) and appending the encapsulation header to the encrypted data packet to create a modified data packet (402, 404, 406), the encapsulation header including an internetwork broadcast address associated with at least one of the first and second networks (N1, N2), whereby the modified data packet (402, 404, 406), has the encapsulation header (430, 432, 434) and an encrypted section that includes encrypted versions of the original header section and the original data section.

2. A method as recited in claim 1 wherein the encapsulation header includes the internetwork broadcast address of the first computer network (N1) as the source identifier for the source host computer (A).

3. A method as recited in claim 1 or 2 wherein the encapsulation header (430, 432, 434) includes the internetwork broadcast address of the second computer network (N2) as a destination identifier for the destination host computer (B).

4. A method as recited in any preceding claim wherein the source identifier in the original header section (420) is the address of the source host computer (A).

5. A method as recited in any preceding claim wherein the destination identifier in the original header section (420) is the address of the destination host computer (B).

6. A method as recited in any preceding claim further comprising:

   receiving the modified data packet (402, 404, 406) at a second bridge (TB2);
   determining in the second bridge (TB2) that the data packet is encrypted upon reference to at least one of the source and destination identifiers in the encapsulation header (432, 434, 436); and
   decrypting the data packet in the second bridge (TB2) to produce a decrypted data packet, wherein after decryption the decrypted original header section is used as the header for the decrypted data packet; and
   forwarding the decrypted packet to the destination host computer (B).

7. A method as recited in any preceding claim wherein the encapsulation header (432,434,436) further includes key management information (440) that identifies the encryption mechanism used to encrypt the data packet.

8. A method as recited in any preceding claim wherein the determination that the data packet (400) should be encrypted is based at least in part upon reference to a data structure that identifies sources and/or destinations for which security is required.

9. A method of decrypting data packets, comprising:

   receiving an encrypted data packet (400) from a source (A) for a destination (B);
   determining that the data packet (400)is encrypted; and
   decrypting the data packet (400) to produce a decrypted data packet,

characterised in that:

the encrypted data packet is provided with an encapsulation header (430, 432, 434) storing a source identifier and a destination identifier, at least one of which is a broadcast address;
the determining step that the data packet is encrypted is carried out upon reference to at least one of the source and destination identifiers;
the encrypted data packet includes an encrypted header section (420) and an encrypted data section (410); and
after decryption the decrypted header section (420) is used as the header for the decrypted data packet (400).

10. A method as recited in claim 9 wherein the receiving, determining and encrypting steps are all performed by a bridge (TB2) associated with a network (N2) that includes the destination,(B) the method further comprising forwarding the decrypted data packet (400) to the destination.

11. A method as recited in claim 9 or claim 10 wherein the encapsulation header (430, 432, 434) further includes key management information that identifies the encryption mechanism used to encrypt the data packet.

12. A bridge (TB 1) for use in a computer network (N 1), the bridge being arranged to route packets (400) transmitted from and received by the network (N1) and to automatically encrypt selected packets (400) passing through the bridge (TB 1), the packets (400) having a header section (420) and a data section (410), the bridge comprising:

an encryption mechanism for encrypting selected packets (400);
means for determining whether a selected packet (400) passing through a bridge is to be encrypted based at least in part on at least one of a source and a destination identified in the header section (420) of the selected packet (400);

characterized by:

the encryption mechanism being arranged to encrypt an entire packet (400) including both the header section (420) and the data section (410); and
means for appending an encapsulation header (430, 432, 434) to the encrypted packets, the encapsulation header (430, 432, 434) having a broadcast address for at least one of a source identifier and a destination identifier for the packet (400), wherein the broadcast address utilized in the encapsulation header is different than the correspondence source or destination identifier in the selected packet (400).

13. A bridge as recited in claim 12 further comprising security information stored in memory in the bridge (TB 1) that identifies specific sources and/or destinations for which security is required, the security information being used to determine whether a selected packet (400) is to be encrypted.

14. A bridge as recited in claim 12 or 13 further comprising:

means for determining whether a received data packet (400) is encrypted upon reference to at least one of the source and destination identifiers in the encapsulation header (430, 432, 434);
a decryption mechanism arranged to decrypt a received packet; and
means for forwarding a decrypted packet to a computing node (B) based on a decrypted destination address.

15. A bridge as recited in any of claims 12 to 14 further comprising means for inserting key management information into the encapsulation header that identify the encryption method used to encrypt the packet (400).

16. A bridge as recited in any of claims 12 to 15 further including a table that identifies source and destination addresses that require encryption.

17. A first network (N1) having a plurality of computing nodes and a bridge (TB1) as recited in any of claims 12 to 16.

18. A system comprising a first network as recited in claim 17 and a second network (N2) having a second bridge (TB2) and a second plurality of computing nodes, wherein the second bridge (TB2) includes:

means for determining whether a received data packet (430, 432, 434, 400) is encrypted upon reference to

at least one of the source and destination identifiers in the encapsulation header;
a decryption mechanism arranged to decrypt the received packet; and
means for forwarding the decrypted packet (400) to a computing node in the second network (N2) based on a decrypted destination address.

19. A bridge (TB2) for use in a computer network, the bridge (TB2) being arranged to route packets (400) transmitted from and received by the network and to automatically decrypt selected packages passing through the bridge, **characterised in that** the bridge is adapted to decrypt data packets (400) having an encapsulation header section (430, 432, 434) and an encrypted section that includes an encrypted header (420) and an encrypted data section (410); and **in that** the bridge further comprises:

means for determining whether a selected packet (400) passing through the bridge is to be decrypted based at least in part on at least one of a source or a destination identifier in the encapsulation header section (430, 432, 434) of the selected packet, being a broadcast address;
a decryption mechanism for decrypting packets (400) that are to be decrypted, wherein the decryption mechanism eliminates the encapsulation header (430, 432, 434), thereby forming a decrypted packet that has a header section (420) and a data section (410); and
means for forwarding decrypted packets (400) to the destination address identified in the decrypted packet.

20. A bridge as recited in claim 19 wherein the decryption mechanism is capable of decrypting packets using a plurality of different encryption algorithms, the bridge further comprising means for detecting key information in the encapsulation header (430, 432, 434) to identify the decryption algorithm to be used to decrypt the packet.

Patentansprüche

1. Verfahren zum Verschlüsseln von Datenpaketen, die über ein Verbindungsnetz von einem Ursprungshostcomputer (A) auf einem ersten Computernetz (N1) mit einer Brücke (TB1) zu einem Zielhostcomputer (B) gesendet wurden, der potentiell Teil eines zweiten Computernetzes (N2) ist, wobei das Verfahren die folgenden Schritte umfasst:

Empfangen eines Datenpakets (400) von dem Ursprungshostcomputer (A) in dem ersten Computernetz (N1) an der Brücke für die Übertragung zum Zielhostcomputer (B),

wobei das Datenpaket (400) einen Originalheaderabschnitt (420) und einen Originaldatenabschnitt (410) beinhaltet, wobei der Originalheaderabschnitt (420) eine den Ursprungshostcomputer (A) identifizierende Ursprungskennung und eine den Zielhostcomputer identifizierende Zielkennung beinhaltet; Ermitteln in der Brücke (TB1), wenigstens teilweise auf der Basis der Bezugnahme auf die Ursprungs- und/oder die Zielkennung, ob das Datenpaket (400) verschlüsselt werden soll; und Verschlüsseln des Datenpakets (400) zum Erzeugen eines verschlüsselten Datenpakets;
wobei das Verfahren **dadurch gekennzeichnet ist, dass**:

die Verschlüsselung des Datenpakets die Verschlüsselung sowohl des Originaldatenabschnitts (410) als auch das Originalheaderabschnitts (420) beinhaltet;
es den Schritt des Erzeugens eines Encapsulation-Headers (430, 432, 434) und Anhängens des Encapsulation-Headers an das verschlüsselte Datenpaket beinhaltet, um ein modifiziertes Datenpaket (402, 404, 406) zu erzeugen, wobei der Encapsulation-Header eine Verbindungsnetzrundsendeadresse beinhaltet, die mit dem ersten und/oder dem zweiten Netzwerk (N1, N2) assoziiert ist, so dass das modifizierte Datenpaket (402, 404, 406) den Encapsulation-Header (430, 432, 434) und einen verschlüsselten Abschnitt hat, der verschlüsselte Versionen des Originalheaderabschnitts und des Originaldatenabschnitts beinhaltet.

2. Verfahren nach Anspruch 1, wobei der Encapsulation-Header die Verbindungsnetzrundsendeadresse des ersten Computernetzes (N1) als Ursprungskennung für den Ursprungshostcomputer (A) beinhaltet.

3. Verfahren nach Anspruch 1 oder 2, wobei der Encapsulation-Header (430, 432, 434) die Verbindungsnetzrundsendeadresse des zweiten Computernetzes (N2) als Zielkennung für den Zielhostcomputer (B) beinhaltet.

4. Verfahren nach einem der vorherigen Ansprüche, wobei die Ursprungskennung im Originalheaderabschnitt (420) die Adresse des Ursprungshostcomputers (A) ist.

5. Verfahren nach einem der vorherigen Ansprüche, wobei die Zielkennung im Originalheaderabschnitt (420) die Adresse des Zielhostcomputers (B) ist.

6. Verfahren nach einem der vorherigen Ansprüche, ferner umfassend:

Empfangen des modifizierten Datenpakets (402, 404, 406) an einer zweiten Brücke (TB2);
Ermitteln in der zweiten Brücke (TB2), nach Bezugnahme auf die Ursprungsund/oder Zielkennung im Encapsulation-Header (432, 434, 436), ob das Datenpaket verschlüsselt ist; und
Entschlüsseln des Datenpakets in der zweiten Brücke (TB2), um ein entschlüsseltes Datenpaket zu erzeugen, wobei der entschlüsselte Originalheaderabschnitt nach dem Entschlüsseln als Header für das entschlüsselte Datenpaket verwendet wird; und
Weiterleiten des entschlüsselten Pakets zum Zielhostcomputer (B).

7. Verfahren nach einem der vorherigen Ansprüche, wobei der Encapsulation-Header (432,434,436) ferner Key-Management-Informationen (440) beinhaltet, die den Verschlüsselungsmechanismus identifizieren, der zum Verschlüsseln des Datenpakets verwendet wurde.

8. Verfahren nach einem der vorherigen Ansprüche, wobei die Ermittlung, dass das Datenpaket (400) verschlüsselt werden soll, wenigstens teilweise auf einer Bezugnahme auf eine Datenstruktur basiert, die Ursprünge und/oder Ziele identifiziert, für die Sicherheit erforderlich ist.

9. Verfahren zum Entschlüsseln von Datenpaketen, umfassend die folgenden Schritte:

Empfangen eines verschlüsselten Datenpakets (400) von einem Ursprung (A) für ein Ziel (B);
Ermitteln, ob das Datenpaket (400) verschlüsselt ist; und
Entschlüsseln des Datenpakets (400) zum Erzeugen eines entschlüsselten Datenpakets,

dadurch gekennzeichnet, dass:

dem verschlüsselten Datenpaket ein Encapsulation-Header (430, 432, 434) gegeben wird, der eine Ursprungskennung und eine Zielkennung speichert, von denen wenigstens eine eine Rundsendeadresse ist;
der Schritt des Ermittelns, ob das Datenpaket verschlüsselt ist, nach Bezugnahme auf die Ursprungs- und/oder Zielkennung erfolgt;
das verschlüsselte Datenpaket einen verschlüsselten Headerabschnitt (420) und einen verschlüsselten Datenabschnitt (410) beinhaltet; und
der entschlüsselte Headerabschnitt (420) nach dem Entschlüsseln als Header für das entschlüsselte Datenpaket (400) verwendet wird.

10. Verfahren nach Anspruch 9, wobei die Empfangs-, Ermittlungs- und Verschlüsselungsschritte alle von einer Brücke (TB2) ausgeführt werden, die mit einem Netzwerk (N2) assoziiert ist, das das Ziel (B) beinhaltet, wobei das Verfahren ferner das Weiterleiten des entschlüsselten Datenpakets (400) zum Ziel umfasst.

11. Verfahren nach Anspruch 9 oder Anspruch 10, wobei der Encapsulation-Header (430, 432, 434) ferner Key-Management-Informationen beinhaltet, die den Verschlüsselungsmechanismus identifizieren, der zum Verschlüsseln des Datenpakets verwendet wurde.

12. Brücke (TB1) für die Verwendung in einem Computemetz (N 1), wobei die Brücke die Aufgabe hat, von dem Netzwerk (N1) gesendete und empfangene Pakete (400) zu leiten und automatisch durch die Brücke (TB 1) passierende selektierte Pakete (400) zu verschlüsseln, wobei die Pakete (400) einen Headerabschnitt (420) und einen Datenabschnitt (410) haben, wobei die Brücke Folgendes umfasst:

einen Verschlüsselungsmechanismus zum Verschlüsseln selektierter Pakete (400);
Mittel zum Ermitteln auf der Basis von wenigstens teilweise einem Ursprung und/oder einem Ziel, der/das im Headerabschnitt (420) des gewählten Pakets (400) identifiziert wurde, ob ein durch eine Brücke passierendes selektiertes Paket (400) verschlüsselt werden soll;

dadurch gekennzeichnet, dass:

der Verschlüsselungsmechanismus die Aufgabe hat, ein gesamtes Paket (400) einschließlich Headerabschnitt (420) und Datenabschnitt (410) zu verschlüsseln; und
Mittel zum Anhängen eines Encapsulation-Headers (430, 432, 434) an die verschlüsselten Pakete, wobei der Encapsulation-Header (430, 432, 434) eine Rundsendeadresse für eine Ursprungskennung und/oder eine Zielkennung für das Paket (400) hat, wobei sich die in dem Encapsulation-Header verwendete Rundsendeadresse von der Korrespondenzursprungs- oder - zielkennung in dem selektierten Paket (400) unterscheidet.

13. Brücke nach Anspruch 12, ferner umfassend Sicherheitsinformationen, die im Speicher in der Brücke (TB1) gespeichert sind und spezifische Ursprünge und/oder Ziele identifizieren, für die Sicherheit erforderlich ist, wobei die Sicherheitsinformationen zum Ermitteln verwendet werden, ob ein selektiertes Paket (400) verschlüsselt werden soll.

14. Brücke nach Anspruch 12 oder 13, die ferner Folgendes umfasst:

Mittel zum Ermitteln, nach Bezugnahme auf die Ursprungs- und/oder Zielkennung in dem Encapsulation-Header (430, 432, 434), ob ein empfangenes Datenpaket (400) verschlüsselt ist;
einen Entschlüsselungsmechanismus zum Entschlüsseln eines empfangenen Paketes; und
Mittel zum Weiterleiten eines entschlüsselten Paketes zu einem Rechenknoten (B) auf der Basis einer entschlüsselten Zieladresse.

15. Brücke nach einem der Ansprüche 12 bis 14, ferner umfassend Mittel zum Einsetzen von Key-Management-Informationen in den Encapsulation-Header, die das zum Verschlüsseln des Pakets (400) verwendete Verschlüsselungsverfahren identifizieren.

16. Brücke nach einem der Ansprüche 12 bis 15, die ferner eine Tabelle beinhaltet, die Ursprungs- und Zieladressen identifiziert, die verschlüsselt werden müssen.

17. Erstes Netzwerk (N1) mit einer Mehrzahl von Rechenknoten und einer Brücke (TB1) nach einem der Ansprüche 12 bis 16.

18. System, umfassend ein erstes Netzwerk nach Anspruch 17 und ein zweites Netzwerk (N2) mit einer zweiten Brücke (TB2) und einer zweiten Mehrzahl von Rechenknoten, wobei die zweite Brücke (TB2) Folgendes umfasst:

Mittel zum Ermitteln nach Bezugnahme auf die Ursprungs- und/oder Zielkennung in dem Encapsulation-Header, ob ein empfangenes Datenpaket (430, 432, 434, 400) verschlüsselt ist;
einen Entschlüsselungsmechanismus zum Entschlüsseln des empfangenen Pakets; und
Mittel zum Weiterleiten des entschlüsselten Pakets (400) zu einem Rechenknoten in dem zweiten Netzwerk (N2) auf der Basis einer entschlüsselten Zieladresse.

19. Brücke (TB2) für die Verwendung in einem Computernetz, wobei die Brücke (TB2) die Aufgabe hat, von dem Netzwerk gesendete und empfangene Pakete (400) zu leiten und automatisch durch die Brücke passierende selektierte Pakete zu entschlüsseln, **dadurch gekennzeichnet, dass** die Brücke die Aufgabe hat, Datenpakete (400) mit einem Encapsulation-Headerabschnitt (430,432,434) und einem verschlüsselten Abschnitt zu entschlüsseln, der einen verschlüsselten Header (420) und einen verschlüsselten Datenabschnitt (410) beinhaltet; und dadurch, dass die Brücke ferner Folgendes umfasst:

Mittel zum Ermitteln, ob ein durch die Brücke passierendes selektiertes Paket (400) entschlüsselt werden soll, auf der Basis von wenigstens einer Ursprungs- und/oder einer Zielkennung in dem Encapsulation-Headerabschnitt (43 0, 432, 434) des selektierten Pakets, die eine Rundsendeadresse ist;
einen Entschlüsselungsmechanismus zum Entschlüsseln von Paketen (400), die entschlüsselt werden müssen, wobei der Entschlüsselungsmechanismus den Encapsulation-Header (430, 432, 434) eliminiert und so ein entschlüsseltes Paket bildet, das einen Headerabschnitt (420) und einen Datenabschnitt (410) hat; und
Mittel zum Weiterleiten von entschlüsselten Paketen (400) zur Zieladresse, die in dem entschlüsselten Paket identifiziert ist.

20. Brücke nach Anspruch 19, wobei der Entschlüsselungsmechanismus Pakete mit einer Mehrzahl von verschiedenen Verschlüsselungsalgorithmen entschlüsseln kann, wobei die Brücke ferner Mittel zum Erfassen von Key-Informationen in dem Encapsulation-Header (430, 432, 434) umfasst, um den zum Entschlüsseln des Pakets zu

verwendenden Entschlüsselungsalgorithmus zu identifizieren.

Revendications

1. Procédé de cryptage de paquets de données transmis par l'intermédiaire d'un inter-réseau depuis un ordinateur hôte source (A) sur un premier réseau informatique (N1) ayant un pont (TB1) jusqu'à un ordinateur hôte destinataire (B) qui fait potentiellement partie d'un deuxième réseau informatique (N2), le procédé comprenant :

   la réception au niveau du pont d'un paquet de données (400) depuis l'ordinateur hôte source (A) dans le premier réseau informatique (N1) pour sa transmission à l'ordinateur hôte destinataire (B), le paquet de données (400) comportant une section d'en-tête d'origine (420) et une section de données d'origine (410), la section d'en-tête d'origine (420) comportant un identifiant de source qui identifie l'ordinateur hôte source (A) et un identifiant de destination qui identifie l'ordinateur hôte destinataire ;
   la détermination dans le pont (TB1) que le paquet de données (400) doit être crypté en fonction au moins en partie d'une référence à au moins un des identifiants de source et de destination ; et
   le cryptage du paquet de données (400) afin de produire un paquet de données crypté ;

   le procédé étant **caractérisé :**

   en ce que le cryptage du paquet de données comporte le cryptage à la fois de la section de données d'origine (410) et de la section d'en-tête d'origine (420) ;
   par l'étape de génération d'un en-tête d'encapsulation (430, 432, 434) et d'ajout de l'en-tête d'encapsulation au paquet de données crypté afin de créer un paquet de données modifié (402, 404, 406), l'en-tête d'encapsulation comportant une adresse de diffusion d'inter-réseau associée à au moins un des premier et deuxième réseaux (N1, N2), si bien que le paquet de données modifié (402, 404, 406), comporte l'en-tête d'encapsulation (430, 432, 434) et une section cryptée qui comporte des versions cryptées de la section d'en-tête d'origine et de la section de données d'origine.

2. Procédé selon la revendication 1, dans lequel l'en-tête d'encapsulation comporte l'adresse de diffusion d'inter-réseau du premier réseau informatique (N1) comme identifiant de source de l'ordinateur hôte source (A).

3. Procédé selon la revendication 1 ou 2, dans lequel l'en-tête d'encapsulation (430, 432, 434) comporte l'adresse de diffusion d'inter-réseau du deuxième réseau informatique (N2) comme identifiant de destination de l'ordinateur hôte destinataire (B).

4. Procédé selon l'une quelconque des revendications précédentes, dans lequel l'identifiant de source dans la section d'en-tête d'origine (420) est l'adresse de l'ordinateur hôte source (A).

5. Procédé selon l'une quelconque des revendications précédentes, dans lequel l'identifiant de destination dans la section d'en-tête d'origine (420) est l'adresse de l'ordinateur hôte destinataire (B).

6. Procédé selon l'une quelconque des revendications précédentes, comprenant en outre :

   la réception du paquet de données modifié (402, 404, 406) au niveau d'un deuxième pont (TB2) ;
   la détermination dans le deuxième pont (TB2) que le paquet de données est crypté en référence à au moins un des identifiants de source et de destination dans l'en-tête d'encapsulation (432, 434, 436) ; et
   le décryptage du paquet de données dans le deuxième pont (TB2) afin de produire un paquet de données décrypté, dans lequel après le décryptage la section d'en-tête d'origine décryptée est utilisée comme en-tête du paquet de données décrypté ; et
   l'envoi du paquet décrypté à l'ordinateur hôte destinataire (B).

7. Procédé selon l'une quelconque des revendications précédentes, dans lequel l'en-tête d'encapsulation (432, 434, 436) comporte en outre des informations de gestion de clé (440) qui identifient le mécanisme de cryptage utilisé pour crypter le paquet de données.

8. Procédé selon l'une quelconque des revendications précédentes, dans lequel la détermination que le paquet de données (400) doit être crypté est basée au moins en partie en référence à une structure de données qui identifie

des sources et/ou destinations devant être sécurisées.

9. Procédé de décryptage de paquets de données, comprenant :

la réception d'un paquet de données crypté (400) depuis une source (A) pour une destination (B) ;
la détermination que le paquet de données (400) est crypté ; et
le décryptage du paquet de données (400) afin de produire un paquet de données décrypté,

caractérisé en ce que :

le paquet de données crypté est fourni avec un en-tête d'encapsulation (430, 432, 434) mémorisant un identifiant de source et un identifiant de destination, dont au moins un est une adresse de destination ;
l'étape de détermination que le paquet de données est crypté est effectuée en référence à au moins un des identifiants de source et de destination ;
le paquet de données crypté comporte une section d'en-tête cryptée (420) et une section de données cryptée (410) ; et
après le décryptage la section d'en-tête décryptée (420) est utilisée comme en-tête du paquet de données décrypté (400).

10. Procédé selon la revendication 9, dans lequel les étapes de réception, détermination et cryptage sont toutes effectuées par un pont (TB2) associé à un réseau (N2) qui comporte la destination (B), le procédé comprenant en outre l'envoi du paquet de données décrypté (400) à la destination.

11. Procédé selon la revendication 9 ou la revendication 10, dans lequel l'en-tête d'encapsulation (430, 432, 434) comporte en outre des informations de gestion de clé qui identifient le mécanisme de cryptage utilisé pour crypter le paquet de données.

12. Pont (TB1) destiné à être utilisé dans un réseau informatique (N1), le pont étant agencé pour router des paquets (400) transmis depuis et reçus par le réseau (N1) et pour crypter automatiquement des paquets sélectionnés (400) passant par le pont (TB1), les paquets (400) ayant une section d'en-tête (420) et une section de données (410), le pont comprenant :

un mécanisme de cryptage pour crypter des paquets sélectionnés (400) ;
un moyen pour déterminer si un paquet sélectionné (400) passant par un pont doit être crypté en fonction au moins en partie d'une source et d'une destination identifiées dans la section d'en-tête (420) du paquet sélectionné (400) ;

caractérisé par :

le mécanisme de cryptage étant agencé pour crypter un paquet complet (400) comportant à la fois la section d'en-tête (420) et la section de données (410) ; et
un moyen pour ajouter un en-tête d'encapsulation (430, 432, 434) aux paquets cryptés, l'en-tête d'encapsulation (430, 432, 434) ayant une adresse de diffusion pour au moins un d'un identifiant de source et d'un identifiant de destination du paquet (400), dans lequel l'adresse de diffusion utilisée dans l'en-tête d'encapsulation est différente de l'identifiant de source ou de destination de correspondance dans le paquet sélectionné (400).

13. Pont selon la revendication 12, comprenant en outre des informations de sécurité mémorisées dans le pont (TB1) qui identifient des sources et/ou destinations spécifiques devant être sécurisées, les informations de sécurité étant utilisées pour déterminer si un paquet sélectionné (400) doit être crypté.

14. Pont selon la revendication 12 ou 13, comprenant en outre :

un moyen pour déterminer si un paquet de données reçu (400) est crypté en référence à au moins un des identifiants de source et de destination dans l'en-tête d'encapsulation (430, 432, 434) ;
un mécanisme de décryptage agencé pour décrypter un paquet reçu ; et
un moyen pour envoyer un paquet décrypté à un noeud informatique (B) basé sur une adresse de destination décryptée.

15. Pont selon l'une quelconque des revendications 12 à 14, comprenant en outre un moyen pour insérer des informations de gestion de clé dans l'en-tête d'encapsulation qui identifient le procédé de cryptage utilisé pour crypter le paquet (400).

16. Pont selon l'une quelconque des revendications 12 à 15, comportant en outre une table qui identifie les adresses de source et de destination qui doivent être cryptées.

17. Premier réseau (N1) ayant une pluralité de noeuds informatiques et un pont (TB1) selon l'une quelconque des revendications 12 à 16.

18. Système comprenant un premier réseau selon la revendication 17 et un deuxième réseau (N2) ayant un deuxième pont (TB2) et une deuxième pluralité de noeuds informatiques, dans lequel le deuxième pont (TB2) comporte :

un moyen pour déterminer si un paquet de données reçu (430, 432, 434, 400) est crypté en référence à au moins un des identifiants de source et de destination dans l'en-tête d'encapsulation ;
un mécanisme de décryptage agencé pour décrypter le paquet reçu ; et
un moyen pour envoyer le paquet décrypté (400) à un noeud informatique dans le deuxième réseau (N2) en fonction d'une adresse de destination décryptée.

19. Pont (TB2) destiné à être utilisé dans un réseau informatique, le pont (TB2) étant agencé pour router des paquets (400) transmis depuis et reçus par le réseau et pour décrypter automatiquement des paquets sélectionnés passant par le pont, **caractérisé en ce que** le pont est adapté pour décrypter des paquets de données (400) ayant une section d'en-tête d'encapsulation (430, 432, 434) et une section cryptée qui comporte un en-tête crypté (420) et une section de données cryptée (410) ; et **en ce que** le pont comprend en outre :

un mécanisme pour déterminer si un paquet sélectionné (400) passant par le pont doit être décrypté en fonction au moins en partie d'un identifiant de source ou de destination dans la section d'en-tête d'encapsulation (430, 432, 434) du paquet sélectionné, étant une adresse de diffusion ;
un mécanisme de décryptage pour décrypter des paquets (400) à décrypter, dans lequel le mécanisme de décryptage élimine l'en-tête d'encapsulation (430, 432, 434), formant ainsi un paquet décrypté qui a une section d'en-tête (420) et une section de données (410) ; et
un moyen pour envoyer des paquets décryptés (400) à l'adresse de destination identifiée dans le paquet décrypté.

20. Pont selon la revendication 19, dans lequel le mécanisme de décryptage est capable de décrypter des paquets au moyen d'une pluralité d'algorithmes de cryptage différents, le pont comprenant en outre un moyen pour détecter des informations de clé dans l'en-tête d'encapsulation (430, 432, 434) afin d'identifier l'algorithme de décryptage à utiliser pour décrypter le paquet.

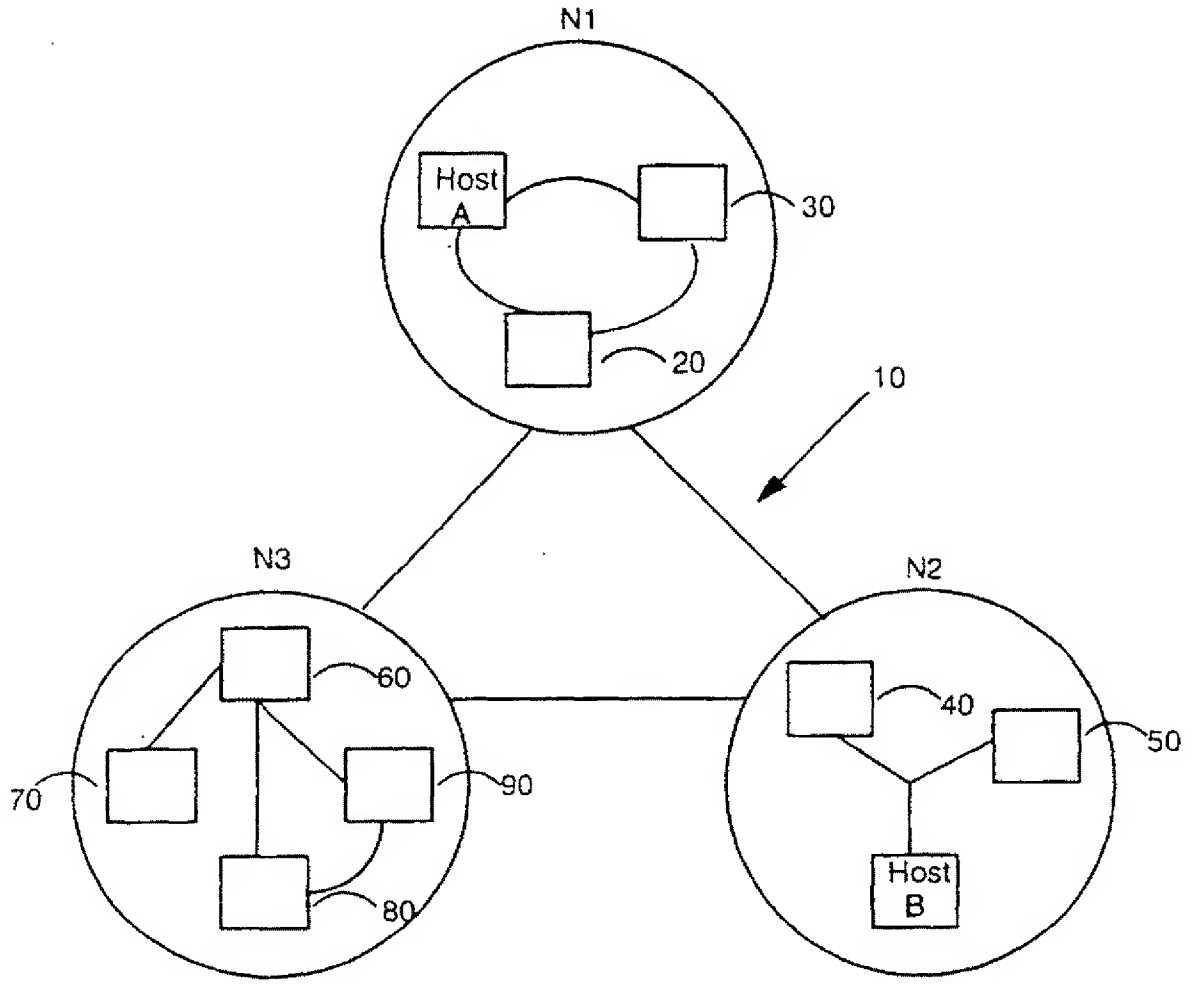


FIG. 1

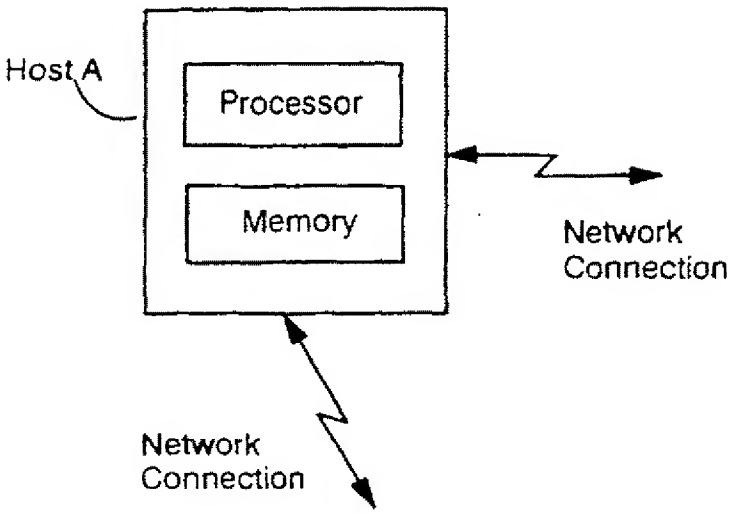

Host A
Processor
Memory
Network
Connection
Network
Connection

FIG. 2

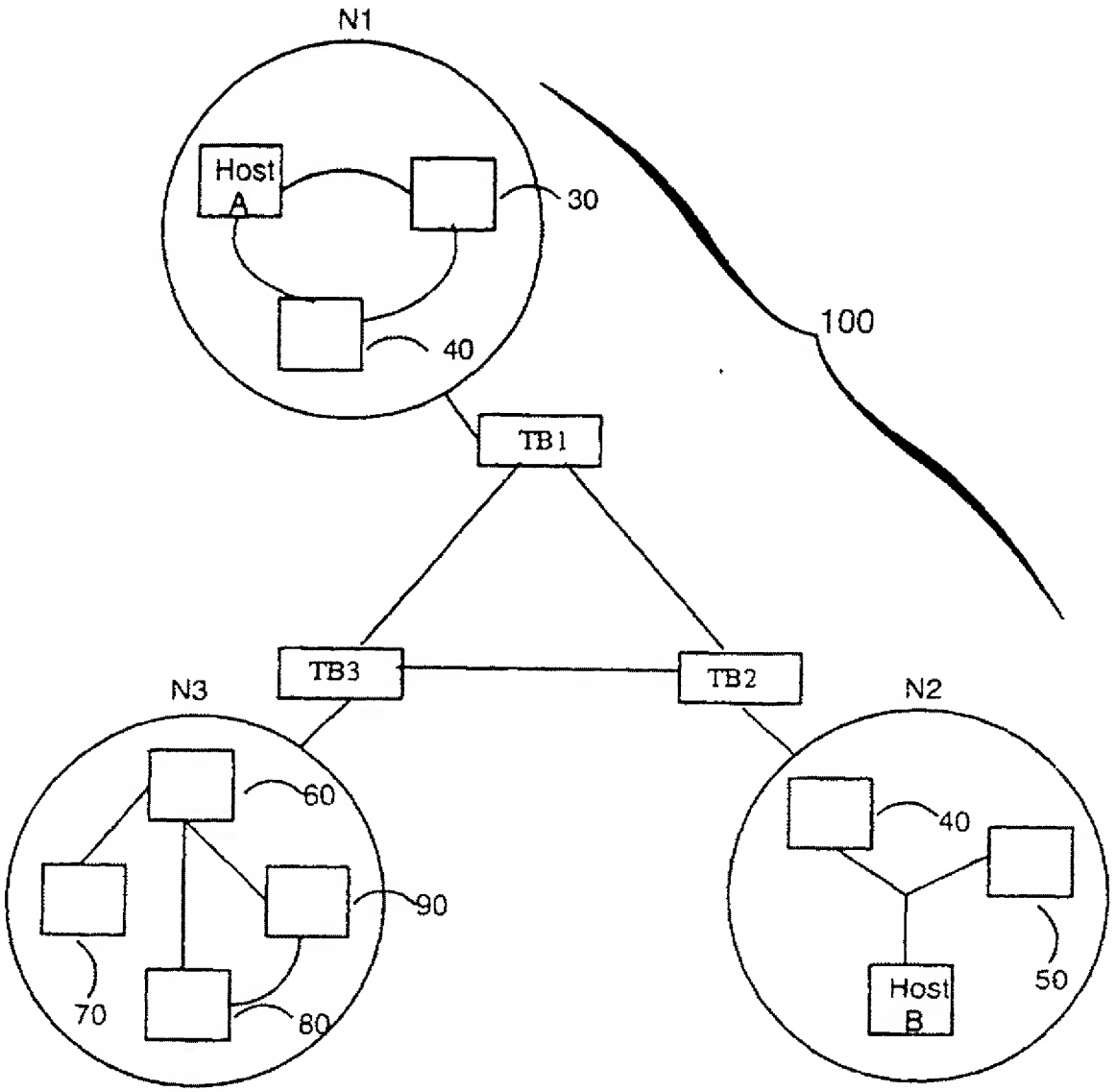


FIG. 3

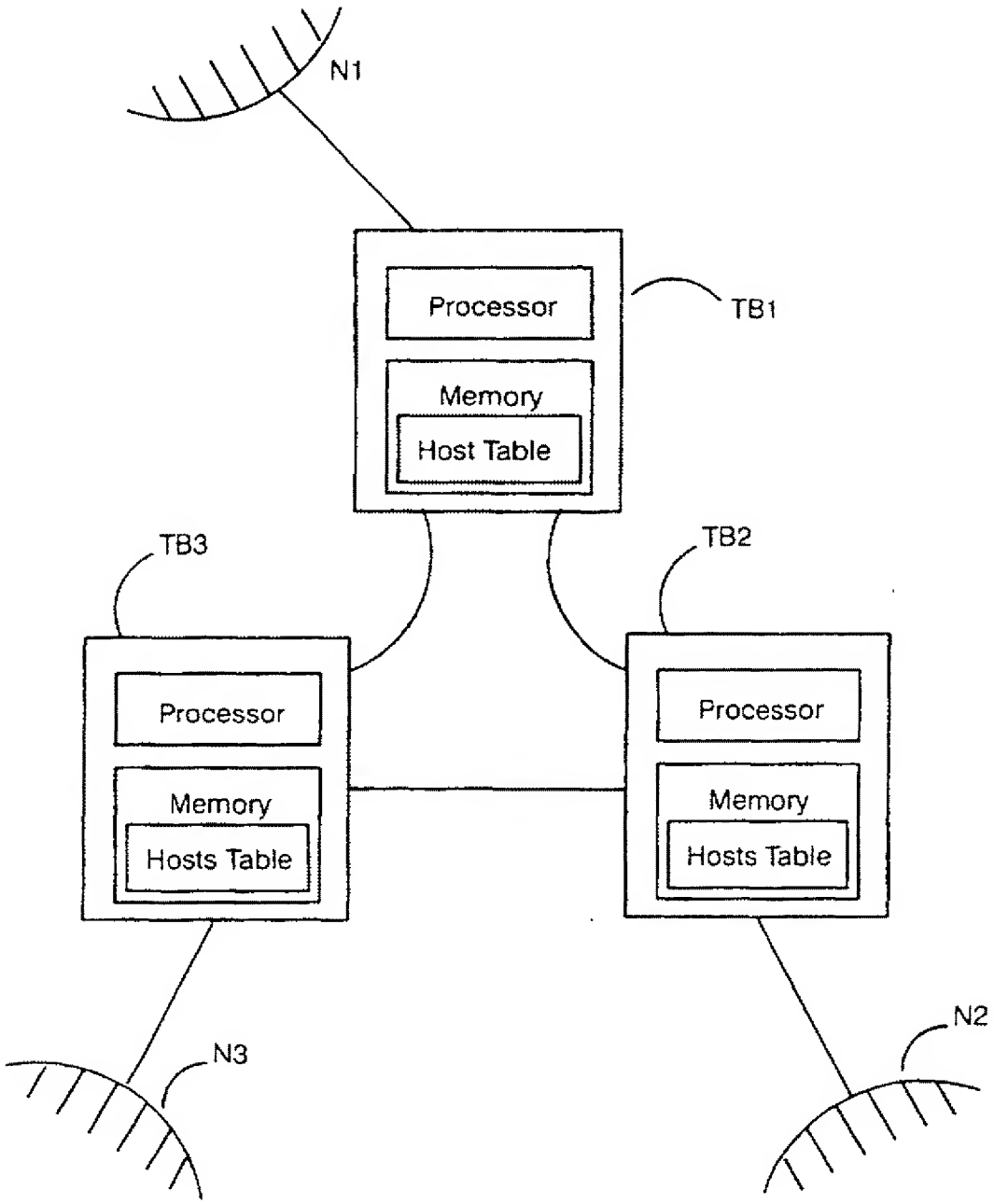

N1
Processor
TB1
Memory
Host Table
TB3
TB2
Processor
Processor
Memory
Memory
Hosts Table
Hosts Table
N3
N2

FIG. 4

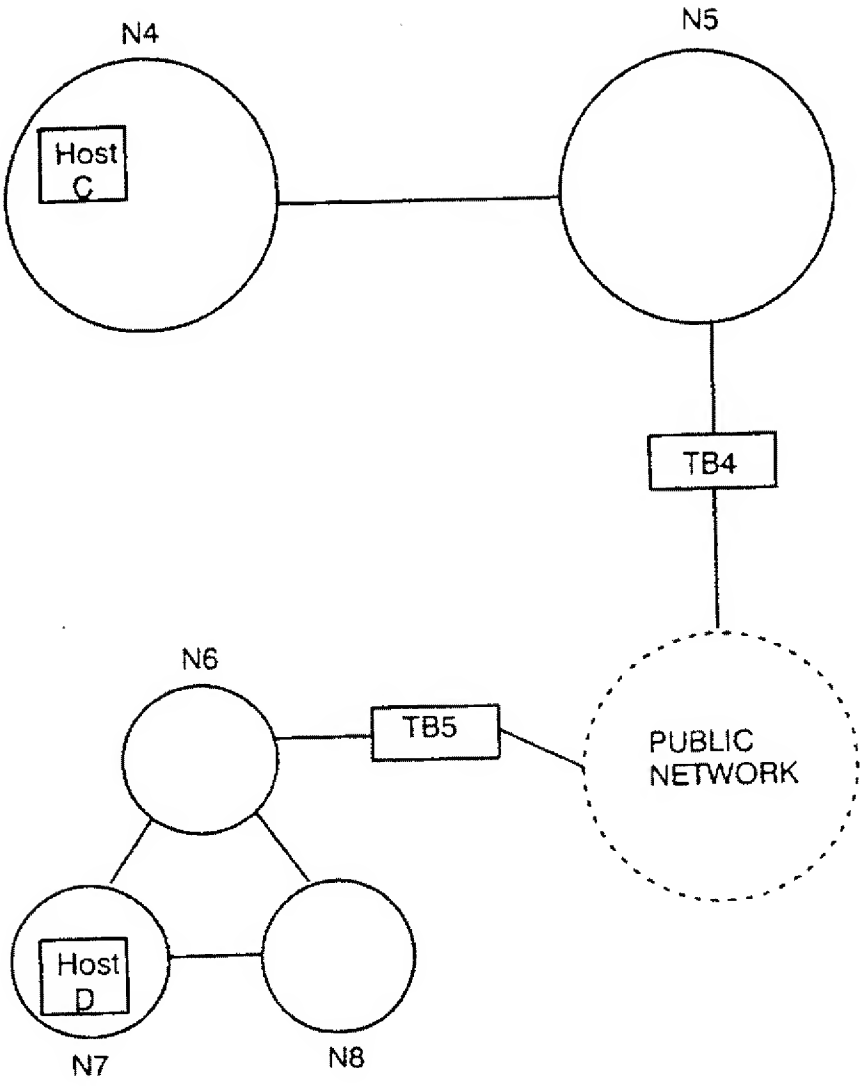


FIG. 5

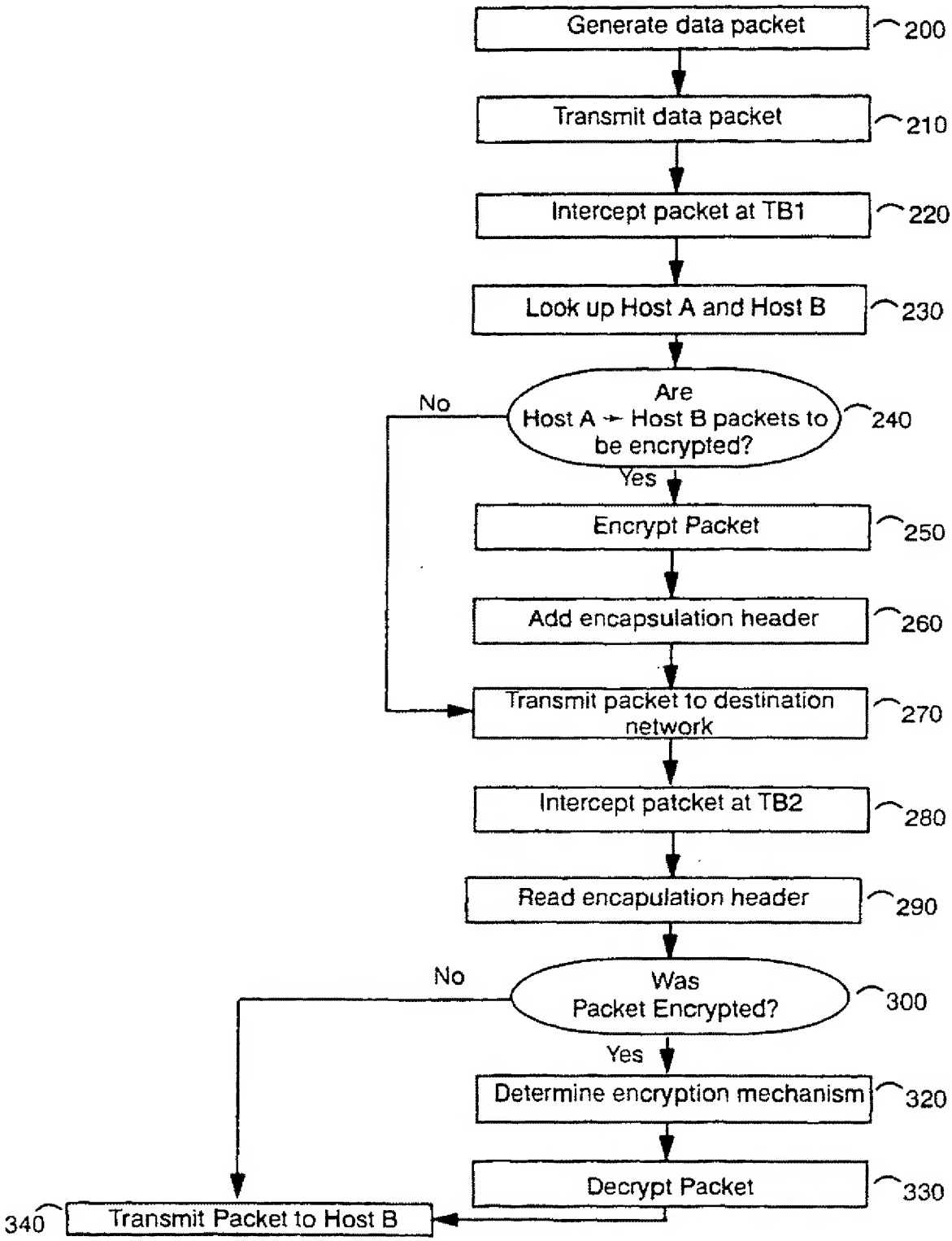

Generate data packet
200
Transmit data packet
210
Intercept packet at TB1
220
Look up Host A and Host B
230
Are Host A ↔ Host B packets to be encrypted?
240
No
Yes
Encrypt Packet
250
Add encapsulation header
260
Transmit packet to destination network
270
Intercept patcket at TB2
280
Read encapulation header
290
Was Packet Encrypted?
300
No
Yes
Determine encryption mechanism
320
Decrypt Packet
330
Transmit Packet to Host B
340

FIG. 6

FIG. 7

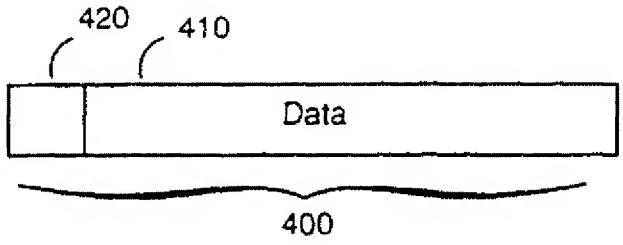


FIG. 8

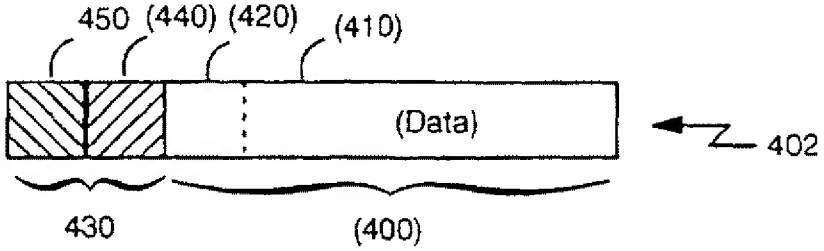


FIG. 9

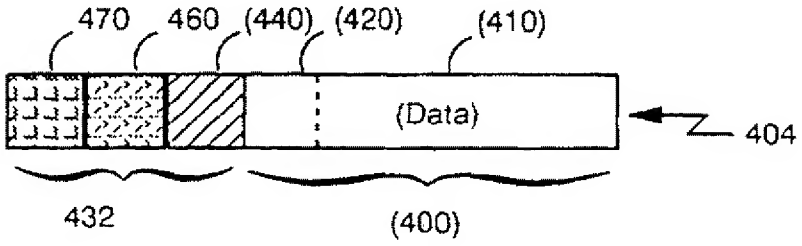
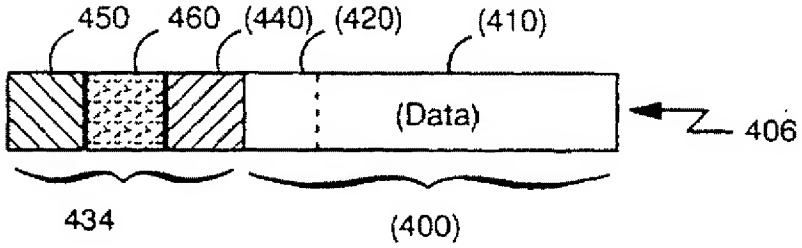


FIG. 10



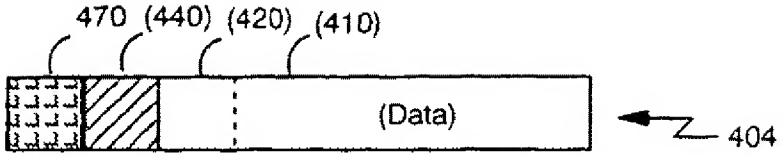


FIG. 11

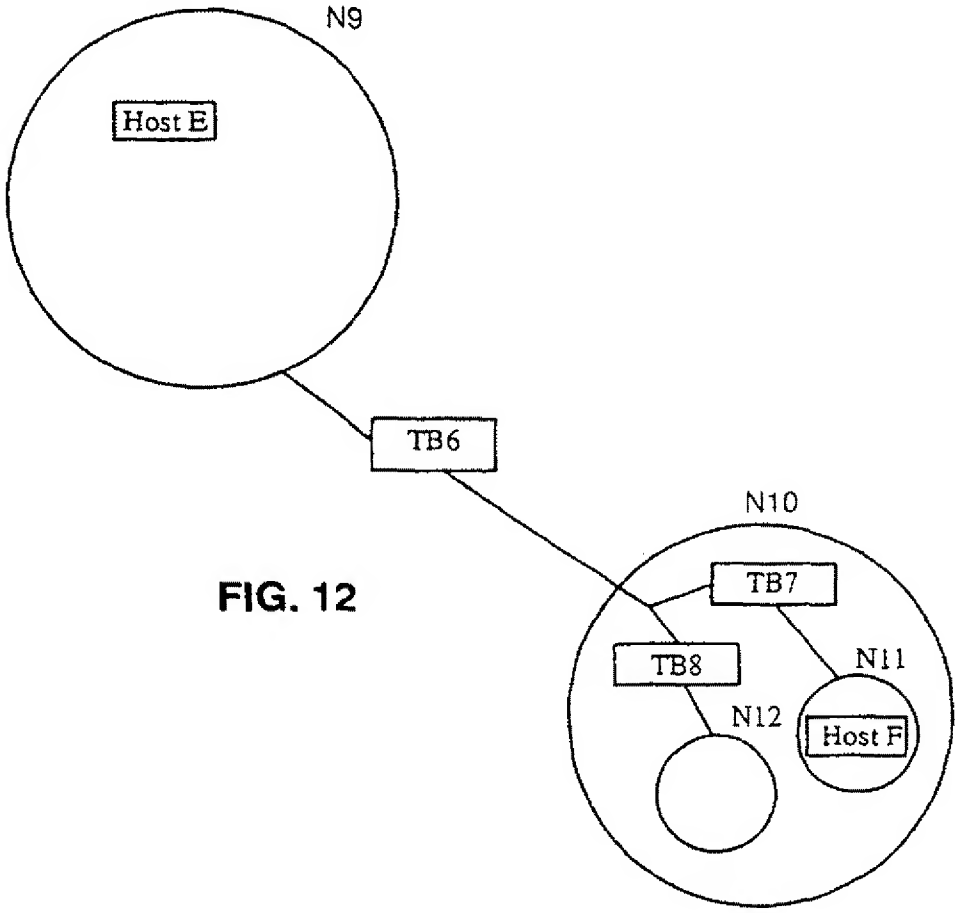


FIG. 12